Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №1

по курсу «Программирование на языке JAVA»

на тему «Графические интерфейсы»

Выполнил: ст. гр. 18ВВ2

Холодков Д.В.

Приняли:

к.т.н., доцент Карамышева Н.С.

к.т.н., доцент Юрова О.В.

Пенза 2022

**Цель работы**

Научиться разрабатывать приложения, обладающие графическим интерфейсом пользователя, с использованием библиотеки Swing.

**Задание**

Вычислить определенный интеграл функции в соответствии с вариантом задания. Разработать приложение, обладающее графическим интерфейсом с использованием языка Java и библиотеки Swing. Приложение должно содержать 3 поля ввода (JTextField), доступных для редактирования, и соответственно таблицу (JTable) с четырьмя колонками: нижняя граница интегрирования, верхняя граница интегрирования, шаг интегрирования и результат вычисления. Кроме того, должны присутствовать 3 кнопки (JButton): добавить, удалить, вычислить. Для добавления/удаления строки и вычисления значения определенного интеграла для функции в соответствии с вариантом задания и параметров выделенной строки таблицы. Результат должен выводиться в четвертой колонке, которая не доступна для редактирования.

Вариант 1. Функция

**Ход работы**

Приложение реализует графический интерфейс и является событийно-управляемым. Окно располагает тремя полями ввода, тремя кнопками и таблицей. В таблице поля “Low”, “Up”, “Step” являются редактируемыми, “Result ” для редактирования недоступно. Окно приложения представлено на рисунке 1.

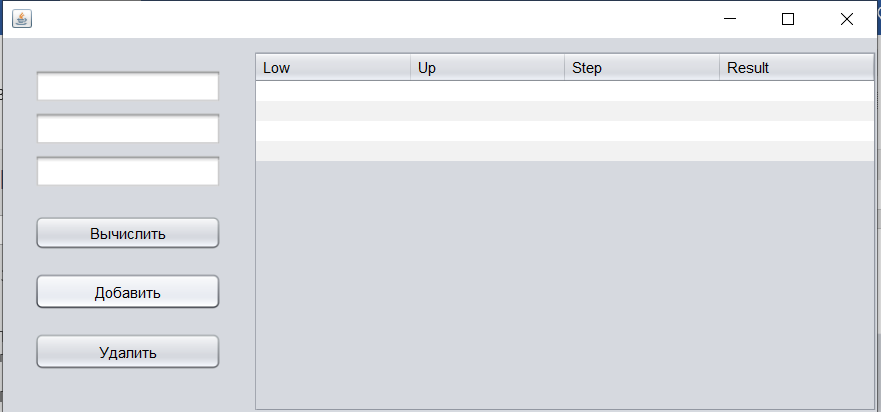


Рисунок 1 – Окно приложения

По нажатию на кнопку “Compute” выполняется метод private void computeButtonMouseClicked(java.awt.event.MouseEvent evt), его алгоритм приведен на рисунке 2.



Рисунок 2 – Схема алгоритма метода computeButtonMouseClicked

По нажатию на кнопку “Add” выполняется метод private void jButton1MouseClicked(java.awt.event.MouseEvent evt). В таблицу добавляется новая строка, заполненная значениями “0.0”. В дальнейшем пользователь может редактировать поля “Left bound”, "Right bound”, “Step (dx)” и вычислить результат.

По нажатию на кнопку “Remove” выполняется метод private void jButton3MouseClicked(java.awt.event.MouseEvent evt). Выбранная пользователем строка удаляется из таблицы. Без предварительного выбора строки нажатие на кнопку не повлияет на поведение программы.

Метод public double integrateInBounds(double left, double right, double dx) реализует алгоритм, представленный на рисунке 3.



Рисунок 3 – Схема алгоритма метода integrateInBounds

**Тестирование**

Окно приложения после запуска представлено на рисунке 4.

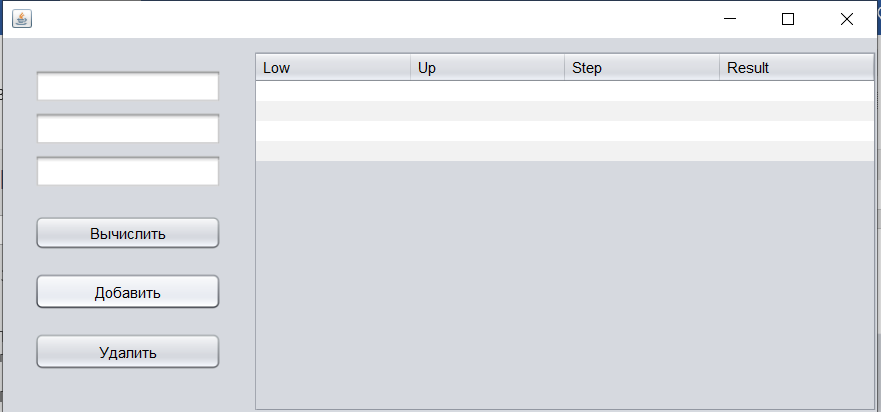


Рисунок 4 – Окно приложения после запуска

Поля содержат некоторые значения по умолчанию чтобы подсказать пользователю формат ввода.

Несколько тестовых значений представлены на рисунке 5.

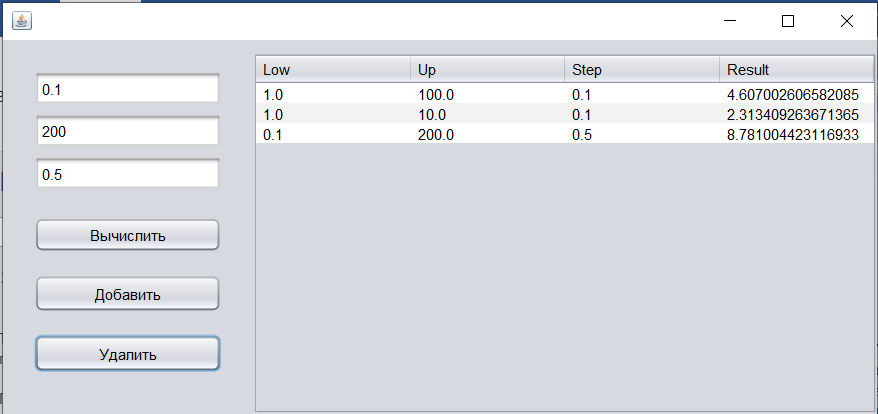
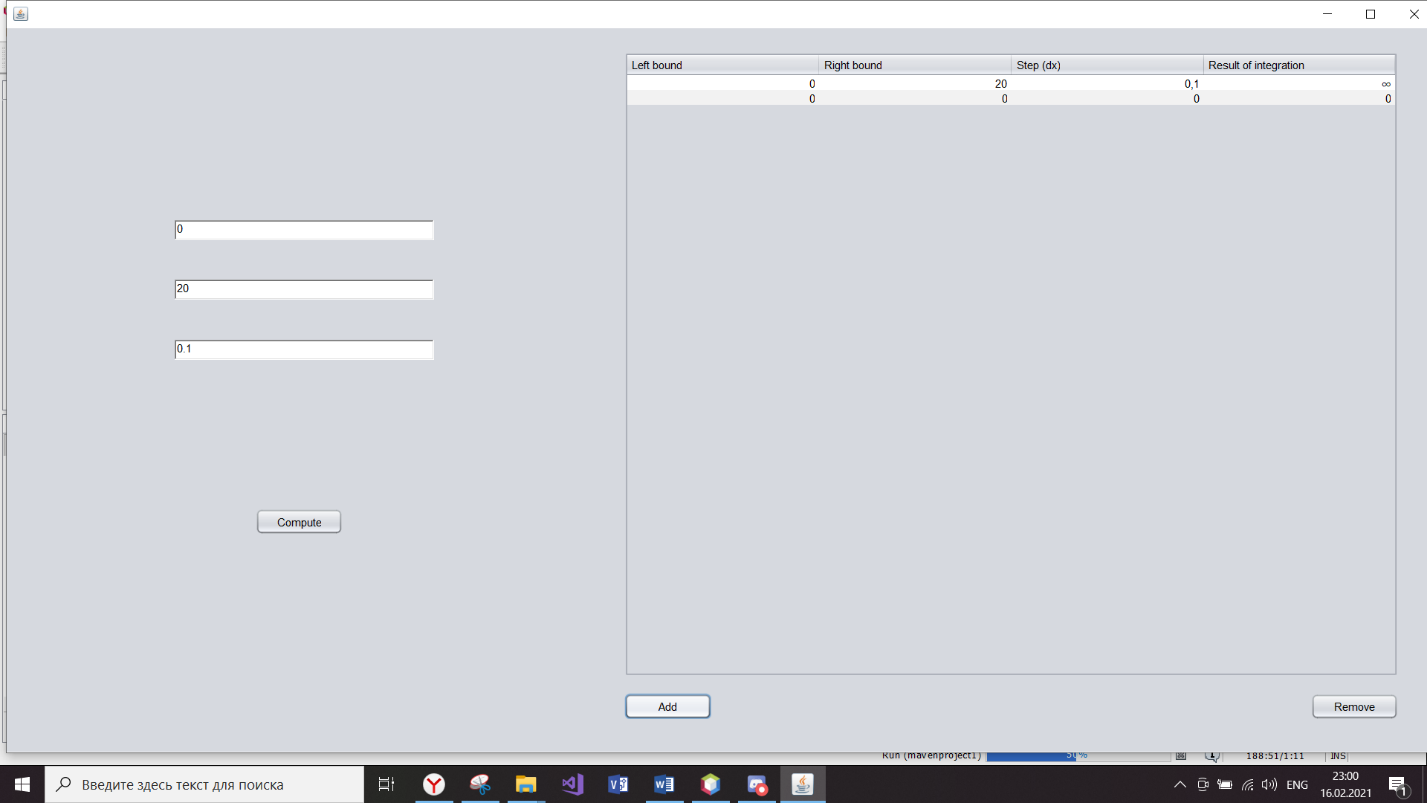


Рисунок 5 – Некоторые тестовые значения

При нажатии на кнопку “Add” выполняется добавление пустой строки, что отражено на рисунке 6.

Рисунок 6 – Добавление пустой строки

После заполнения строки пользователь кликает на кнопку “Compute”, результат выводится в правый столбец. Результат представлен на рисунке 8.

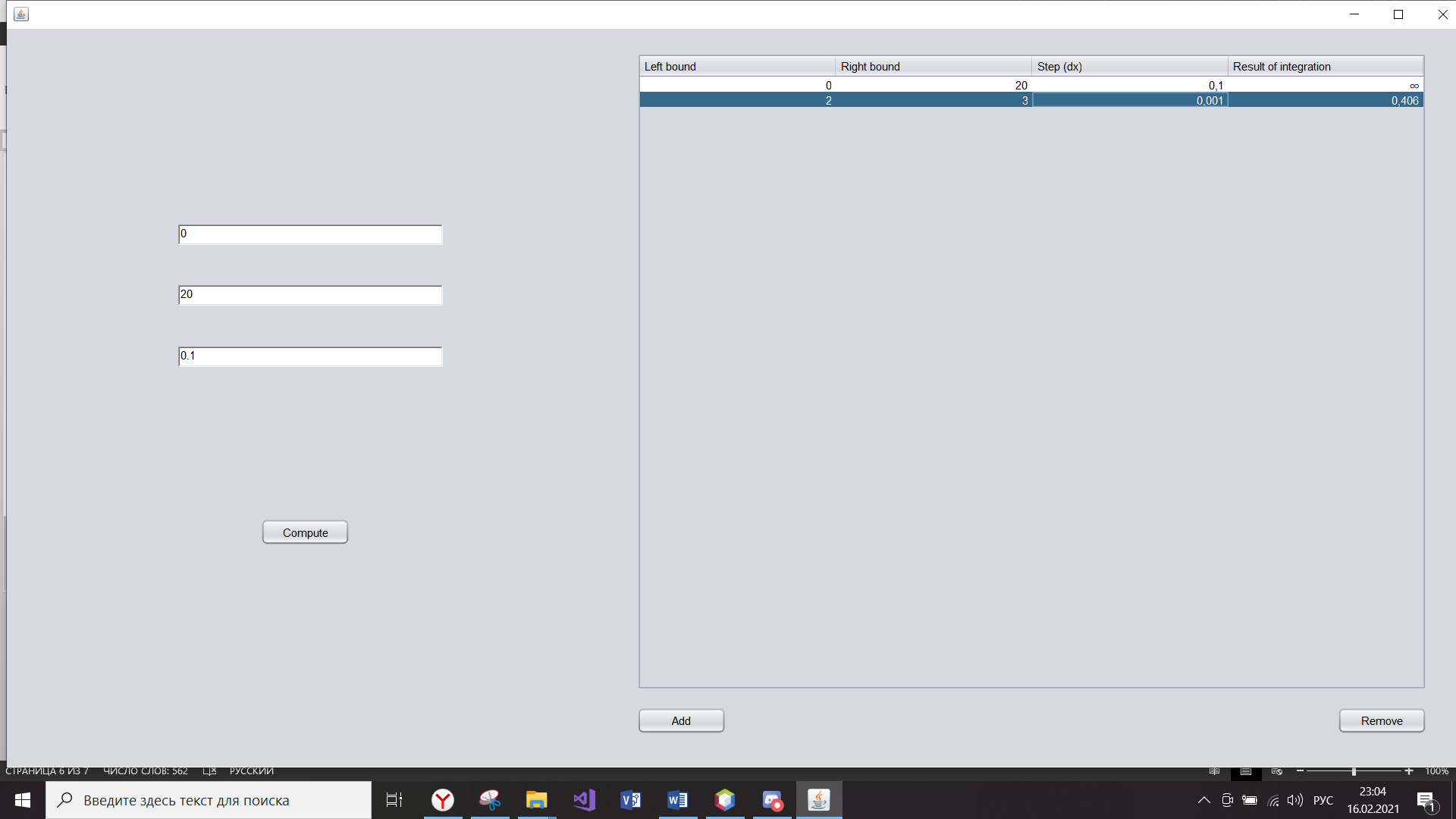


Рисунок 7 – Вывод результата

Пользователь может выбрать строку и удалить ее, нажав на кнопку “Remove”. Результат показан на рисунке 8.

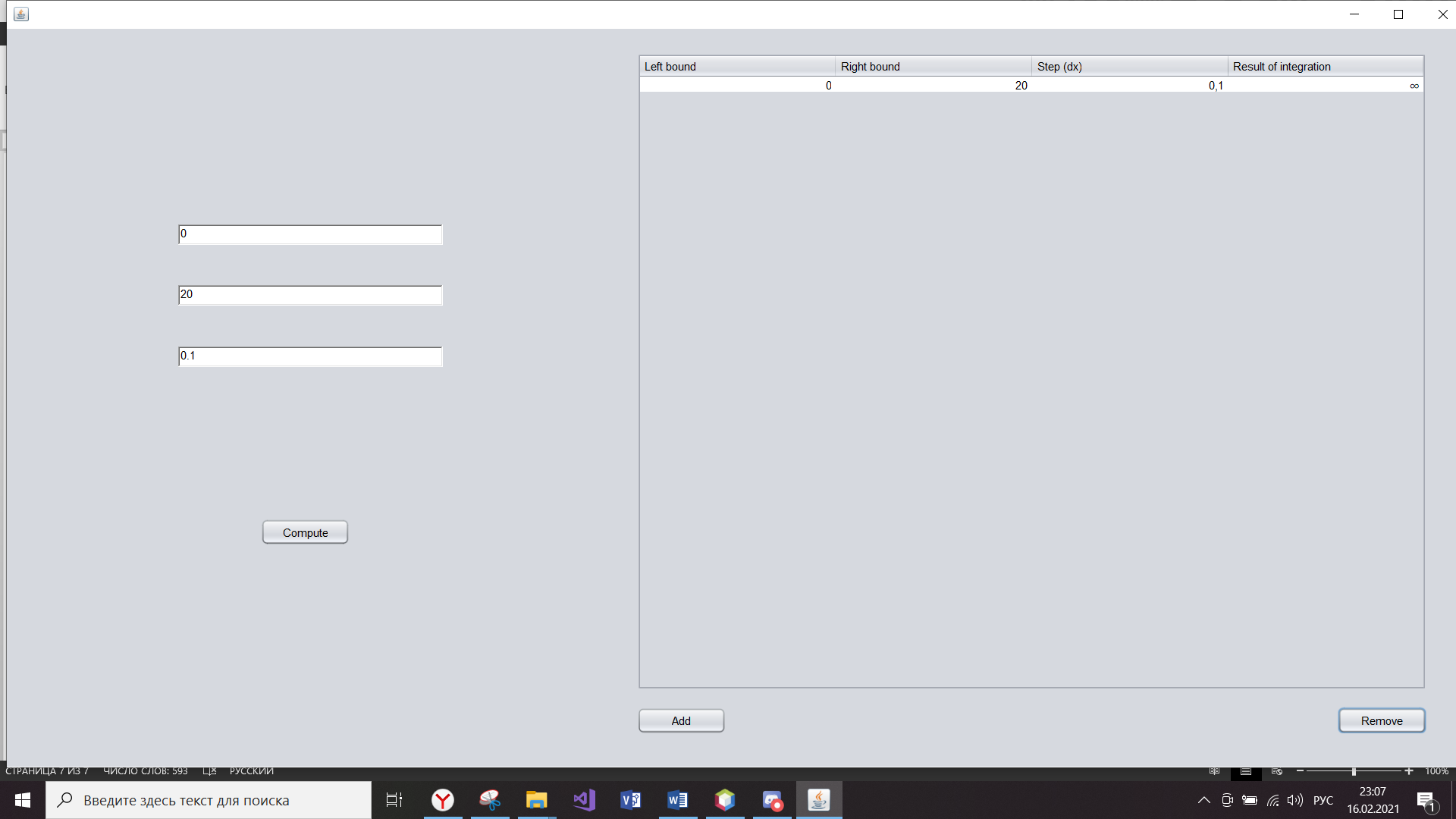


Рисунок 8 – Таблица после удаления строки

Для проверки правильности вычислений был использован онлайн калькулятор определенного интеграла kontrolnaya-rabota.ru.

Тестовый пример 1: . Расчеты интеграла с разными шагами интегрирования представлены на рисунке 9. Контрольный просчет с помощью онлайн калькулятора приведен на рисунке 10.



Рисунок 9 – Расчеты первого тестового примера

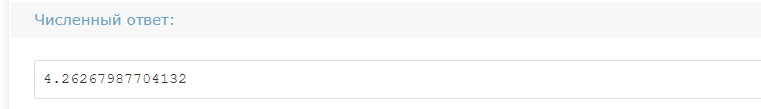


Рисунок 11 – Контрольный просчет первого тестового примера

Тестовый пример 2: . Расчеты интеграла с разными шагами интегрирования представлены на рисунке 12. Контрольный просчет с помощью онлайн калькулятора приведен на рисунке 13.

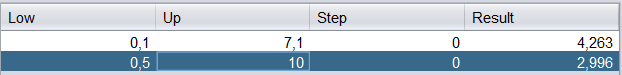


Рисунок 12 – Расчеты второго тестового примера



Рисунок 13 – Контрольный просчет второго тестового примера

**Листинг программы**

Файл IntegralForm.java

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template

\*/

package com.mycompany.lab1;

import java.util.Vector;

import javax.swing.table.DefaultTableModel;

/\*\*

\*

\* @author kekos

\*/

public class Lab1 extends javax.swing.JFrame {

final Integral<Function> integral;

/\*\*

\* Creates new form Lab1

\*/

public Lab1() {

integral = new Integral<>((x)->1/x);

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jTextField1 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

getContentPane().add(jTextField1, new org.netbeans.lib.awtextra.AbsoluteConstraints(25, 25, 150, -1));

getContentPane().add(jTextField2, new org.netbeans.lib.awtextra.AbsoluteConstraints(25, 59, 150, -1));

getContentPane().add(jTextField3, new org.netbeans.lib.awtextra.AbsoluteConstraints(25, 93, 150, -1));

jButton1.setText("Добавить");

jButton1.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jButton1MouseClicked(evt);

}

});

getContentPane().add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(25, 188, 150, 30));

jButton2.setText("Удалить");

jButton2.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jButton2MouseClicked(evt);

}

});

getContentPane().add(jButton2, new org.netbeans.lib.awtextra.AbsoluteConstraints(25, 236, 150, 30));

jButton3.setText("Вычислить");

jButton3.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jButton3MouseClicked(evt);

}

});

jButton3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton3ActionPerformed(evt);

}

});

getContentPane().add(jButton3, new org.netbeans.lib.awtextra.AbsoluteConstraints(25, 142, 150, 28));

jTable1.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null},

{null, null, null, null},

{null, null, null, null},

{null, null, null, null}

},

new String [] {

"Low", "Up", "Step", "Result"

}

) {

Class[] types = new Class [] {

java.lang.Double.class, java.lang.Double.class, java.lang.Double.class, java.lang.Double.class

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

});

jScrollPane1.setViewportView(jTable1);

getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 10, 500, 290));

pack();

}// </editor-fold>

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jButton3MouseClicked(java.awt.event.MouseEvent evt) {

int selectedRow = jTable1.getSelectedRow();

double left, rigth, dx;

Vector<Double> row = null;

if(selectedRow != -1){

row = ((DefaultTableModel) jTable1.getModel()).getDataVector().elementAt(selectedRow);

left = row.get(0);

rigth = row.get(1);

dx = row.get(2);

} else {

left = Double.parseDouble(jTextField1.getText());

rigth = Double.parseDouble(jTextField2.getText());

dx = Double.parseDouble(jTextField3.getText());

}

double result = integral.integrateInBounds(left, rigth, dx);

if(row!=null){

row.insertElementAt(result, 3);

}

else {

Double[] arr = {left, rigth, dx, result};

DefaultTableModel model = (DefaultTableModel)jTable1.getModel();

model.addRow(arr);

}

// TODO add your handling code here:

}

private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {

int selectedRow = jTable1.getSelectedRow();

DefaultTableModel model = (DefaultTableModel)jTable1.getModel();

if(selectedRow >= 0){

model.removeRow(selectedRow);

}

// TODO add your handling code here:

}

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {

DefaultTableModel model = (DefaultTableModel)jTable1.getModel();

model.addRow(new Object[]{0.0,0.0,0.0,0.0});

// TODO add your handling code here:

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Lab1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Lab1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Lab1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Lab1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Lab1().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

private javax.swing.JTextField jTextField3;

// End of variables declaration

}

Файл Integral.java

package com.team.mavenproject1;

public class Integral<T extends Function> {

private final T function;

Integral(T function){

this.function = function;

}

public double integrateInBounds(double left, double right, double dx){

double sum = 0.0;

double currentRigth = left + dx;

while(true){

sum += (function.compute(left) + function.compute(currentRigth)) \* 0.5 \* dx;

if(currentRigth >= right){

break;

}

left += dx;

currentRigth = Math.min(currentRigth + dx, right);

}

return sum;

}

public T getFunction(){

return function;

}

}

Файл Function.java

package com.team.mavenproject1;

public interface Function {

double compute(double x);

}

**Вывод**

Научились разрабатывать приложения, обладающие графическим интерфейсом пользователя, с использованием библиотеки Swing.